PACKET SCHEDULING METHOD AND APPARATUS

BACKGROUND OF THE INVENTION

[01] This application claims priority from Chinese Patent Application No. 03118918.0, filed on April 11, 2003, in the Chinese Intellectual Property Office and Korean Patent Application No. 2004-19627, filed on March 23, 2004, in the Korean Intellectual Property Office, the disclosures of which are incorporated herein in their entirety by reference.

1. Field of the Invention

[02] The present invention relates to a network communication system and applications thereof, and more particularly, to a packet scheduling method that can be especially applied to packet scheduling in a router.

2. Description of the Related Art

[03] The development of the Internet has lead to considerable advancements in the multimedia industry. Generally, bandwidth and delay are two factors that largely affect audio and video multimedia applications. As such, a router requires effective and fast packet scheduling to provide reliable quality of service (QoS) to network devices.

[04] In order to perform the function of QoS, a scheduling algorithm based on a generalized processor sharing (GPS) model, such as WFQ, WF^2Q, or WF^2Q^+ has been widely used. The GPS model is an ideal stream model based on the following assumptions: (1) the length of a packet can be unlimitedly divided; and (2) all the streams can accept the service at the same time. Since in a practical system, the minimum unit that a scheduler serves is a packet and the scheduler serves only one stream at one time, it cannot be realized by the GPS model. Jon C.R. Bennett and Hui Zhang (J. Bennett and H. Zhang, "Hierarchical Packet Fair Queuing Algorithms", Proceedings of the ACM-SIGCOMM96, pages 143-156, Palo Alto,

CA, August 1996) have developed a WF^{2}Q scheduling algorithm to simulate the GPS model in a practical system. The basic idea of the algorithm is to maintain a start service time and a finish service time for each packet in the stream. Before the scheduler transmits a packet, it needs to carry out the quality test for the packet which is to be scheduled. Only packets whose start service time is shorter than a system virtual time can pass the test, and the packet with the minimum finish service time in the packets which have passed the test will be sent. This strategy is called the smallest eligible virtual finish time first (SEFF) selection strategy.

[05] Because the WF^{2}Q$^+$ algorithm has good fairness and delay properties and is not extremely complicated, it has been broadly used in the field. However, there are some practical problems when applying it: first, the complexity of the algorithm is increased with the increase of the stream to be scheduled, in particular, in the case of a high-speed core router, if the quantity of a data stream is large, the application of the algorithm will cause much load on the system, and second, it is not easy to realize the hardware to perform the algorithm.

SUMMARY OF THE INVENTION

[06] The present invention provides a packet scheduling method and apparatus, which are simple and effective and guarantee the performance of a WF^{2}Q$^+$ algorithm for quality of service (QoS).

[07] According to an aspect of the present invention, there is provided a packet scheduling method. The packet scheduling method divides scheduled packets into a first stream queue and a second stream queue and stores the packets in the first and second stream queues, and then, performs scheduling on each packet by a smallest eligible virtual finish time first (SEFF) strategy. The procedures are as follows:

[08] (1) A scheduling node is initialized, and an initial value of a system virtual time is set.

[09] (2) If a predetermined packet reaches the scheduling node, it is checked whether the packet is a first packet of a data stream. If the packet is the first packet of the data stream, the packet is stored at an end of a first stream queue Q1 according to a data rate and/or length of a corresponding data stream, and a virtual start service time is counted by Equation 2. The time is a virtual start service time of the data stream. If the packet is not the first packet of the data stream, the packet is directly stored at the end of the data stream.

[10] (3) In scheduling, a scheduler scans the virtual start service time for a head packet of a first data stream in all the queues. After that, a legal packet whose virtual start service time is shorter than the system virtual time is detected, and a virtual finish service time of the legal packet is counted by Equation 4. After that, the packet is transmitted at a minimum virtual finish service time.

[11] (4) The procedures of transmitting the selected head packet are as follows: first, a packet is extracted from a corresponding data stream and transmitted. After that, the packet of the data stream is stored at an end of a backlog stream queue $Q_2(R_x, L_y)$ according to the data rate and length of a new head packet. After that, the virtual start service time of the data stream is renewed by Equation 3 (described later). This time is also a virtual start service time of a new head packet of the data stream. After that, the system virtual time is renewed by Equation 1 (described later).

[12] (5) The above-described procedures (2) through (4) are repeatedly performed until scheduling is terminated.

[13] The packet scheduling method comprises (a) classifying a stream according to a data rate and/or length of a packet; (b) if the packet of the classified stream is a first packet, storing the packet in a first stream queue, and if the packet of the classified stream is a subsequent packet, storing the packet in a second stream queue; (c) counting a virtual start service time of the packet stored in the first stream queue according to a weighted fairness

queuing method; and (d) counting a virtual start service time of the packet stored in the second stream queue as a virtual start service time of the previous packet.

[14] Step (c) is performed in accordance with the following equation;

[15] $S_i^k = \max(V(a_i^k), F_i^{k-1})\,(\text{where } Q_i = 0)$,

[16] where S_i^k is a virtual start service time of a k-th packet of an i-th stream, V(t) is a system virtual time function, a_i^k is an arrival time of the k-th packet of the i-th stream, F_i^{k-1} is a virtual finish service time of a (k-1)-th packet of the i-th stream, and Q_i is the quantity of the previous packet contained in a corresponding queue of the i-th stream.

[17] The method further comprises (e) detecting a legal packet whose virtual start service time is shorter than a system virtual service time by scanning the virtual start service time of the packets stored in the first stream queue and the second stream queue.

[18] According to another aspect of the present invention, there is provided a packet scheduling apparatus. The packet scheduling apparatus comprises a classifier, which classifies a stream according to a data rate and/or length of a packet; a first stream queue in which a first packet of the classified stream is stored; a second stream queue in which a subsequent packet of the classified stream is stored; and a SEFF selector, which detects a legal packet from all the packets stored in the first stream queue and the second stream queue according to a SEFF strategy.

BRIEF DESCRIPTION OF THE DRAWINGS

[19] The above aspects and advantages of the present invention will become more apparent by describing in detail exemplary embodiments thereof with reference to the attached drawings in which:

[20] FIG. 1 schematically shows a packet scheduling method according to the present invention;

[21] FIG. 2 shows an internal structure of first and second stream queues according to an embodiment of the present invention;

[22] FIG. 3 is a time flowchart showing a packet scheduling method according to an embodiment of the present invention;

[23] FIG. 4 is a time flowchart showing a system virtual service time is renewed between transmission of a previous packet and transmission of a next packet;

[24] FIG. 5 shows a structure of first and second stream queues according to another embodiment of the present invention;

[25] FIG. 6 shows a simulation topological structure according to the present Invention;

[26] FIG. 7 shows the bandwidth of a data stream according to the embodiment shown in FIG. 2;

[27] FIG. 8 shows the bandwidth jitter of the data stream according to the embodiment shown in FIG. 2;

[28] FIG. 9 shows the delay of the data stream according to the embodiment shown in FIG. 2;

[29] FIG. 10 shows the delay jitter of a data stream according to the embodiment shown in FIG. 2;

[30] FIG. 11 shows the bandwidth of a data stream according to the embodiment shown in FIG. 5;

[31] FIG. 12 shows the bandwidth jitter of the data stream according to the embodiment shown in FIG. 5;

[32] FIG. 13 shows the delay of the data stream according to the embodiment shown in FIG. 5; and

[33] FIG. 14 shows the delay jitter of a data stream according to the embodiment shown in FIG. 5.

DETAILED DESCRIPTION OF THE INVENTION

[34] Hereinafter, exemplary embodiments of the present invention will be described in detail with reference to the accompanying drawings.

[35] First, equations and symbols that will be used in the following descriptions are defined using Equations 1 through 4 and Table 1.

[36] $V(t+\tau) = \max(V(t)+\tau, \min_{i\in B(t)} S_i^{hi(t)})$ ....(1)

[37] $S_i^k = \max(V(a_i^k), F_i^{k-1})(\text{where } \mathrm{Q_i} = 0)$ ....(2)

[38] $S_i^k = F_i^{k-1}\ (\text{where } \mathrm{Q_i} \neq 0)$ ....(3)

[39] $F_i^k = S_i^k + \dfrac{L_i^k}{R_i(t)}$ ....(4)

Table 1

V(t)	Virtual time function of system
S_i^k	Virtual start service time of packet k of data stream i
F_i^k	Virtual finish service time of packet k of data stream i
τ	Renewal time-interval of system virtual time
B(t)	Set of all the streams to be backlogged in system at time t
Hi(t)	Serial number of head packet of data stream i at time t
Qi	Quantity of packet to be scheduled in data stream i
a_i^k	Arrival time of packet k of data stream i
L_i^k	Length of packet k of data stream i
$R_i(t)$	Data rate of data stream i at time t

[40] FIG. 1 shows a structure of a scheduler for explaining a packet scheduling method according to an embodiment of the present invention.

[41] A scheduler 100 comprises a classifier 130, a first stream queue 110, a second stream queue 120, and a SEFF selector 140.

[42] A virtual time function V(t) of the scheduler 100 is given by Equation 1.

[43] $V(t+\tau) = \max(V(t)+\tau, \min_{i\in B(t)} S_i^{hi(t)}$ ....(1)

[44] V(t) is a virtual time function of the scheduler 100, τ is a time-interval of system virtual time renewal, B(t) is the assembly of all the streams to be backlogged in the scheduler 100, hi(t) is a serial number of a head packet of a data stream i, and S_i^k is a virtual start service time of a k-th packet.

[45] The classifier 130 classifies data streams according to a data rate. A quantification grade is M, which is sequentially marked as R_1, R_2, ... , and R_m. Likewise, packets of the data streams are classified according to lengths by a quantification grade N, which is sequentially marked as L_1, L_2, ... , and L_n. For different data streams, the grade of length quantification may be the same or not as that of other data streams. Various data queues are obtained by a combination of R and L, and each of the data queues is represented as $Q(R_m, L_n)$. As such, M × N data queues are obtained, and a packet 132 which enters into the scheduler 100 is classified according to the rate of a stream to which the packet belongs and the length of the packet and stored in different queues.

[46] For a predetermined data rate ($R=R_m$) and the length L ($L_{n-1}<L\leq L_n$) of a head packet, each data stream is stored in a corresponding queue $Q(R_m, L_n)$. The head packet is stored at presently first positions of the first and second stream queues 110 and 120. In the classifier 130, the grade of length quantification of the head packet is the nearest grade length longer than itself. The length longer than the length of the highest grade is classified as the highest grade length. The data streams in the queue are represented as F_1, F_2, ... , and F_{tail}. The packets in the data stream F_i are represented as P_{i1}, P_{i2}, ... , and P_{itail}.

[47] A first packet of the data stream corresponds to a packet which is processed for the first time of a system or is first processed when the system restarts after it stops for a predetermined amount of time. To verify a new packet, when there is no packet waiting for scheduling in the data stream of the packet, the packet is regarded as a first packet. When there is a packet waiting for scheduling in the data stream of the packet, the packet is called a

subsequent packet of the data stream. Since the count of the virtual start service time of the first packet is not coincident with the count of the subsequent packet of the data stream, the first packet of the data stream should be processed separately. Thus, as shown in FIG. 1, the data queue is divided into two parts, that is, the first stream queue 110 and the second stream queue 120. Processing the first packet of the data stream is performed on the first stream queue 110, and processing the subsequent packet of the data stream is performed on the second stream queue 120. Accordingly, the first and second stream queues 110 and 120 need 2 × M × N queues altogether. The SEFF selector 140 selects packets from the first and second stream queues 110 and 120 according to a SEFF strategy and transmits the selected packets to a next node for a predetermined amount of time.

[48] FIG. 2 shows an internal structure of the first and second stream queues 110 and 120.

[49] The classifier 130 determines in which one of the first and second queues 110 and 120 a packet 132 that enters into the scheduler 100 is stored according to a data rate and the length of the packet. In this case, if the packet is a first packet 112, the packet is stored in a predetermined queue $Q_1(R_1, L_1)$ in the first stream queue 110. If the packet is a subsequent packet 122, the packet is stored in a predetermined queue $Q_2(R_1, L_1)$ in the second stream queue 120. The packets 112 and 122 stored in the queues $Q_1(R_1, L_1)$ and $Q_2(R_1, L_1)$ have the same packet length and stream rate.

[50] FIG. 3 is a time flowchart showing a packet scheduling method according to an embodiment of the present invention.

[51] In step 310, the scheduler 100 is initialized, and an initial value of a virtual start service time of the scheduler 100 is set to 0, for example.

[52] In step 320, if a predetermined packet reaches the scheduler 100, the classifier 130 checks whether the packet is a first packet of the data stream.

[53] In step 330, if the packet is the first packet, the packet is stored in the first stream queue $Q_1(R_m, L_n)$ according to the data rate and/or length of the packet.

[54] In step 340, the virtual start service time is counted by Equation 2. The time is a virtual start service time of a corresponding data stream.

[55] $$S_i^k = \max(V(a_i^k), F_i^{k-1})(\text{where } Q_i = 0) \ldots\ldots (2)$$

[56] If it is determined in step 320 that the entered packet is not the first packet of the corresponding stream, in step 332, the packet is stored directly in the second stream queue 120, and in step 334, the virtual start service time is counted by Equation 3.

[57] $$S_i^k = F_i^{k-1} \ (\text{where } Q_i \neq 0) \ldots\ldots (3)$$

[58] In step 350, the SEFF selector 140 scans a virtual start service time for a head packet of a first stream in all the queues in the first and second stream queues 110 and 120. In this case, all the queues include M × N first stream queues $Q_1(R, L)$ and M × N second stream queues $Q_2(R, L)$.

[59] After that, in step 360, the SEFF selector 140 detects a packet, that is, a legal packet, whose virtual start service time is shorter than a present system virtual time.

[60] In step 370, a virtual finish service time of the legal packet is counted by Equation 4.

[61] $$F_i^k = S_i^k + \frac{L_i^k}{R_i(t)} \ldots\ldots (4)$$

[62] After that, in step 380, the SEFF selector 140 transmits the detected legal packet to a next node at a minimum virtual finish service time. The above-described steps are repeatedly performed until all the packets are transmitted.

[63] FIG. 4 is a time flowchart showing a system virtual service time is renewed between transmission of a previous packet and transmission of a next packet.

[64] When selecting and transmitting a head packet of a first stream of a first stream queue $Q_1(R_m, L_n)$ or a second stream queue $Q_2(R_m, L_n)$, the following procedure is used.

[65] First, in step 410, the SEFF selector 140 extracts a packet from a data stream F and transmits the extracted packet.

[66] Then, in step 420, a new head packet is stored in the first stream queue $Q_1(R_x, L_x)$ or the second stream queue $Q_2(R_x, L_y)$ in which the head packet is selected and transmitted. In addition, a next packet of a data stream F is stored in the second stream queue $Q_2(R_x, L_y)$ according to the data rate and length of the new head packet.

[67] After that, in step 430, the virtual start service time of the data stream F is renewed by Equation 3. The time is also a virtual start service time of a new head packet of the data stream F.

[68] After that, in steps 440 and 450, the system virtual time is renewed by Equation 1.

[69] After that, in step 460, extraction and transmission of the next packet is performed, and then, the above-described steps are repeatedly performed to renew the service time.

[70] In an embodiment of the present invention, a scheduling node used in the packet scheduling method according to the present invention is a router.

[71] FIG. 5 shows a structure of first and second stream queues according to another embodiment of the present invention.

[72] In FIG. 5, the structure of the second stream queue is the same as that of FIG. 2, but the structure of a first stream queue 510 is different from that of FIG. 2. This is because the structure of the first stream queue is simplified to decrease the quantity of the queues and save hardware resources. Referring to FIG. 5, the first stream queue 510 is classified into either the length of a packet or a data rate and is marked by $Q_1(L_n)$ or $Q_1(R_m)$. Thus, the queue in the first stream queue 510 needs $M \times N + N$ or $M \times N + M$ queues.

[73] Hereinafter, a packet scheduling method using queues by length quantification will be described with reference to FIG. 5.

[74] (4) A scheduling node is initialized, and an initial value of a system virtual time is set to 0, for example.

[75] (5) If a predetermined packet reaches the scheduling node, it is checked whether the packet is a first packet of a data stream. If the packet is the first packet of the data stream, the packet is stored at an end of a first stream queue $Q_1(L_n)$ corresponding to a data rate of the packet, and a virtual start service time is counted by Equation 2. The time is a virtual start service time of a corresponding data stream. If the packet is not the first packet of the data stream, the packet is directly stored at the end of the corresponding data stream.

[76] (6) In scheduling, a scheduler scans the virtual start service time for a head packet of a first stream in all the queues. In this case, all the queues include N first stream queues $Q_1(L)$ and $M \times N$ second stream queues $Q_2(R, L)$. After that, a legal packet whose virtual start service time is shorter than the system virtual time is extracted, and a virtual finish service time of the legal packet is counted by Equation 4. After that, the corresponding packet is transmitted at a minimum virtual finish service time.

[77] (4) When selecting and transmitting a head packet of a first stream of a first stream queue $Q_1(L_n)$ or a second stream queue $Q_2(R_m, L_n)$, the following procedure is commonly used. First, a packet is extracted from a corresponding data stream and transmitted. After that, the packet of the data stream is stored at an end of the second stream queue $Q_2(R_x, L_y)$ according to the data rate and length of a new head packet. After that, the virtual start service time of the data stream is renewed by Equation 3. This time is also a virtual start service time of a new head packet of the data stream. After that, the system virtual time is renewed by Equation 1.

[78] (6) The above-described procedures (2) through (4) are repeatedly performed until scheduling is terminated.

[79] FIG. 6 shows a simulation topological structure according to the present invention.

[80] In FIG. 6, the bandwidth of each input chain is 10 M. All the data streams are scheduled by a scheduling node and are output through an output chain. The rate and length of the scheduling node shown in FIG. 6 are classified into five grades using a simplified method shown in FIG. 5.

[81] Five rate grades for the data stream used in this simulation are as follows:

[82] 0.1 Mbps, 0.3 Mbps, 1 Mbps, 2 Mbps, and 5 Mbps.

[83] Five length grades for the data stream used in this simulation are as follows:

[84] 200 bytes, 400 bytes, 800 bytes, 1,000 bytes, and 1,600 bytes.

[85] Transmission methods for the data streams used in this simulation are as follows and an On/Off method has been used in this simulation:

[86] - constant bit rate (CBR): indicates that the data stream is transmitted at a constant rate.

[87] - On/Off: indicates that the data stream is transmitted intermittently.

[88] - the same length: the length of the packets in the data stream is the same.

[89] - average distribution: the length of the packets in the data stream is evenly distributed within a predetermined range.

[90] - normal distribution: the length of the packets in the data stream is normally distributed in a predetermined range based on a predetermined central value.

[91] FIGS. 7 through 14 show simulation results of scheduling according to the present invention. To verify the performance of the packet scheduling method according to the present invention, simulation was performed using a network simulator (NS), and the following performance indices were tested.

[92] * bandwidth: practical bandwidth Mbps received by each data stream.

[93] * bandwidth jitter: average practical difference Mbps of a bandwidth received by each data stream at an adjacent period.

[94] * delay: difference, measured in ms, between start and arrival times of the packets of each data stream;

[95] * delay jitter: average delay difference between previous packet and subsequent packet of each data stream;

[96] Results refer to part of a detailed performance test method. In general, the method is simple and effective and realizes hardware easily. In addition, the present invention guarantees the performance of a WF^2Q^+ algorithm.

[97] Parameters used in this simulation are shown in Table 2.

Table 2

Data Stream mark	Chain bandwidth	Performed bandwidth		Transmission method and rate of data stream		Length of packets
1	10 Mbps	5.0 Mbps	50%	On/Off	5.0 Mbps	Normal distribution
2	10 Mbps	2.0 Mbps	20%	On/Off	2.0 Mbps	Normal distribution
3	10 Mbps	1.0 Mbps	10%	On/Off	1.0 Mbps	Normal distribution
4	10 Mbps	1.0 Mbps	10%	On/Off	1.0 Mbps	Normal distribution
5	10 Mbps	0.3 Mbps	3%	On/Off	0.3 Mbps	Normal distribution
6	10 Mbps	0.3 Mbps	3%	On/Off	0.3 Mbps	Normal distribution
7	10 Mbps	0.1 Mbps	1%	On/Off	0.1 Mbps	Normal distribution
8	10 Mbps	0.1 Mbps	1%	On/Off	0.1 Mbps	Normal distribution
9	10 Mbps	0.1 Mbps	1%	On/Off	0.1 Mbps	Normal distribution
10	10 Mbps	0.1 Mbps	1%	On/Off	0.1 Mbps	Normal distribution
Scheduling Output	10Mbps					

[98] An On/Off model has been used in each data stream of the above configuration. The packet in the data stream uses normal distribution (the average value is 1,000 bits and the deviation is 400 bits). In this test, the performance indices, such as bandwidth, bandwidth jitter, delay, and delay jitter, in the two methods shown in FIGS. 2 and 5 have been measured.

[99] FIGS. 7 through 10 show bandwidth, bandwidth jitter, delay, and delay jitter according to the method shown in FIG. 2. FIGS. 11 through 14 show bandwidth, bandwidth jitter, delay, and delay jitter according to the method shown in FIG. 5.

[100] According to simulation results of FIGS. 7 through 14, the two scheduling methods according to the present invention guarantee the four performance indices. Thus, clients' quality of service (QoS) can be guaranteed.

[101] Based on a simulation performed using a NS, field programmable gate array (FPGA) of Xilinx company has been used to realize scheduling chips. The chips support the maximum of 128k data streams, five rate grades and five length grades. In addition, the chips can configure a variety of values of all the rates and length grades. According to a practical processing test, the chips can guarantee the performed bandwidth, delay, and fairness of each data stream.

[102] Meanwhile, the packet scheduling method according to the present invention can be written as computer programs. Codes, and code segments of the programs can be easily construed by programmers skilled in the art to which the present invention pertains. In addition, the programs are stored in a computer readable medium, are read and performed by a computer, thereby implementing packet scheduling. Examples of the computer readable recording medium include magnetic storage media, optical recording media, and storage media such as carrier waves.

[103] As described above, according to the present invention, even though the number of data streams is increased, packet scheduling using a simple structure of hardware can be performed.

[104] While this invention has been particularly shown and described with reference to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The preferred embodiments should be considered in descriptive sense only and not for purposes of limitation. Therefore, the scope of the invention is defined not by the detailed description of the invention but by the

appended claims, and all differences within the scope will be construed as being included in the present invention.